OUTPUT STABILIZABILITY OF DISCRETE EVENT DYNAMIC SYSTEMS[1]

Cüneyt M. Özveren[2]

Alan S. Willsky[2]

June 21, 1989

Abstract

In this paper, we investigate the problem of designing stabilizing feedback compensators for Discrete Event Dynamic Systems (DEDS). The DEDS model used is a finite-state automaton in which some transition events are controllable and some events are observed. The problem of output stabilization is defined as the construction of a compensator such that the closed loop system is stable, in the sense that all state trajectories go through a given set E infinitely often. We define a stronger notion of output stabilizability which requires that we also have perfect knowledge of the state in E through which the trajectory passes on each of its visits to E. Necessary and sufficient conditions are presented for both notions. The complexity of these tests is polynomial in the cardinality of the state space of the observer. A number of sufficient conditions for the weaker notion are also presented. Corresponding tests for these sufficient conditions are shown to be polynomial in the cardinality of the state space of the system. Finally, a problem of resilient output stabilizability is addressed.

[1]Research supported by the Air Force Office of Scientific Research under Grant AFOSR-88-0032 and by the Army Research Office under Grant DAAL03-86-K0171.

[2]Laboratory for Information and Decision Systems, MIT, Cambridge, MA 02139.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
21 JUN 1989		**00-06-1989 to 00-06-1989**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Output Stabilizability of Discrete Event Dynamic Systems	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Massachusetts Institute of Technology,Laboratory for Information and Decision Systems,77 Massachusetts Avenue,Cambridge,MA,02139-4307**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**49**	

1 Introduction

Discrete Event Dynamic Systems (DEDS) are dynamic systems, for which the evolution of the state is triggered by the instantaneous occurence of discrete events. Such behavior can be found in many complex, man-made systems at some level of abstraction, such as flexible manufacturing systems and communication systems. Although DEDS have been studied extensively by computer scientists, the notion of control of a DEDS has been introduced only recently, by Wonham, Ramadge, et al. [3,7,8,10]. This work assumes a finite state model and that certain events in the system can be enabled or disabled. The control of the system is achieved by choice of control inputs that enable or disable these events. The objective is to control the system, so that the event trajectory in this system is always in a given set of desired strings of events. This approach is generally classified as a linguistic approach, since the objective is defined in terms of the language generated by the closed-loop system, i.e., the set of possible strings of events. This work was extended by Cieslak et al. [1] and Lin and Wonham [2] for the case of partial event observations. However, as shown by Tsitsiklis in [9], most partial observation problems of interest are NP-hard, in the cardinality of the state space of the system.

The work of Wonham et al. has prompted a considerable response by other researchers in the field, and one of the principal characteristics of this research has been the exploration of alternate formulations and paradigms that provide the opportunity for new and important developments building on the foundations of both computer science and control. The work presented here is very much in that spirit with, perhaps, closer ties to more standard control concepts. In particular, in our work, we have had in mind the development of the elements needed for a regulator

theory for DEDS. In another paper, [5], we develop notions of stability and stabilizability for DEDS which might, more correctly, be thought of as properties of resiliency or error-recovery. In [4], we focus on the questions of observability and state reconstruction. We assume what might be thought of as an intermittent observation model: no direct measurements of the state are made, and we only observe a specified subset of possible events, i.e., if an event outside this subset occurs, we will not observe it and indeed will not even know that an event has occurred. We also define a notion of resiliency which allows us to characterize resilient observers which generate correct estimates in a finite number of transitions following a burst of measurement errors. In this paper, we combine our work on stabilizability and observability to address a problem of stabilization by dynamic output feedback under partial observations. Specifically, we construct stabilizing compensators by cascading an observer and a stabilizing full-state feedback defined on the state space of the observer. While this is a well-established control-theoretic approach, there are several important distinguishing features of the DEDS compensation problem. First of all, in the context of linear systems, we know that observability together with stabilization by state feedback imply the existence of and provide the basis for designing stabilizing output compensators. Thanks to the intermittent nature of observations, the same is not true for the class of DEDS considered in this paper. Secondly, since the observers we construct for DEDS keep track of all possible states in which the DEDS can be, it is possible to re-cast the output stabilization problem as the stabilization of the observer by state feedback. Finally, a critical issue of particular importance in the DEDS context is computational, and thus it is essential that one characterizes the complexity in designing and implementing a stabilizing compensator.

In the next section, we introduce the mathematical framework considered in this paper and summarize our previous work. In Section 3, we formulate two notions of output stabilization and present algorithms for constructing compensators for both problems of output stabilization in polynomial time in the cardinality of the state space of the observer. In Section 4, we present sufficient conditions for output stabilizability that can be tested in polynomial time in the cardinality of the state space of the sytem. In Section 5, we present our treatment of the problem of resilient output stabilization. Finally, in Section 6, we summarize our results and discuss several directions for further work.

2 Background and Preliminaries

2.1 System Model

The class of systems we consider are nondeterministic finite-state automata with intermittent event observations. The basic object of interest is the quadruple:

$$G = (X, \Sigma, \Gamma, U) \tag{2.1}$$

where X is the finite set of states, with $n = |X|$, Σ is the finite set of possible events, $\Gamma \subset \Sigma$ is the set of observable events, and U is the set of admissible control inputs consisting of a specified collection of subsets of Σ, corresponding to the choices of sets of controllable events that can be enabled. The dynamics defined on G that we consider in [5] are of the form:

$$x[k+1] \in f(x[k], \sigma[k+1]) \tag{2.2}$$

$$\sigma[k+1] \in (d(x[k]) \cap u[k]) \cup e(x[k]) \tag{2.3}$$

Here, $x[k] \in X$ is the state after the kth event, $\sigma[k] \in \Sigma$ is the $(k+1)$st event, and $u[k] \in U$ is the control input after the kth event. The function $d : X \to 2^{\Sigma}$ is a set-valued function that specifies the set of possible events defined at each state (so that, in general, not all events are possible from each state), $e : X \to 2^{\Sigma}$ is a set valued function that specifies the set of events that cannot be disabled at each state, and the function $f : X \times \Sigma \to X$ is also set-valued, so that the state following a particular event is not necessarily known with certainty. Without loss of generality, we assume that $e(x) \subset d(x)$ for all x. The set $d(x)$ represents an "upper bound" on the set of events that can occur at state x, whereas the set $e(x)$, is a lower bound. The effect of our control action is adjusting the set of possible events between these bounds,

by disabling some of the controllable events, i.e., elements of the set $d(x) \cap \overline{e(x)}$. Note that in this general framework, there is no loss of generality in taking $U = 2^{\Sigma}$. Also, by appropriate choice of $e(x)$, we can model situations in which we have enabling/disabling control over some events only at certain states. In Section 4, we will use this general framework. Up to that point however, we assume the slightly more restrictive framework of [8] in which there is an event subset $\Phi \subset \Sigma$ such that we have complete control over events in Φ and no control over events in $\overline{\Phi}$, the complement of Φ. In this case, we can take $U = 2^{\Phi}$ and

$$e(x) = d(x[k]) \cap \overline{\Phi} \tag{2.4}$$

Furthermore, we assume that $\Phi \subset \Gamma$. These assumptions simplify the presentation of our results, but it is possible to get similar results, at a cost of additional computational complexity, if our assumptions on controllable events are relaxed.

Our model of the output process is quite simple: whenever an event in Γ occurs, we observe it; otherwise, we see nothing. Specifically, we define the output function $h : \Sigma \to \Gamma \cup \{\epsilon\}$, where ϵ is the "null transition", by

$$h(\sigma) = \begin{cases} \sigma & \text{if } \sigma \in \Gamma \\ \epsilon & \text{otherwise} \end{cases} \tag{2.5}$$

Then, our output equation is

$$\gamma[k+1] = h(\sigma[k+1]) \tag{2.6}$$

Note that h can be thought of as a map from Σ^* to Γ^*, where Γ^* denotes the set of all strings of finite length with elements in Γ, including the empty string ϵ. In particular, $h(\sigma_1 \cdots \sigma_n) = h(\sigma_1) \cdots h(\sigma_n)$. The quadruple $A = (G, f, d, h)$ representing our system can also be visualized graphically as in Figure 2.1. Here, circles denote

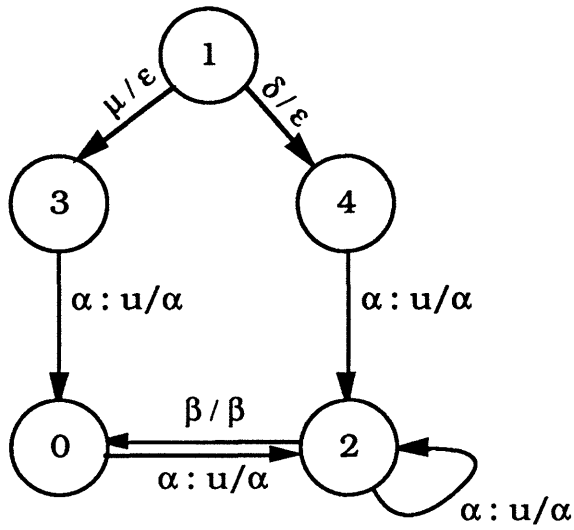

Figure 2.1: A Simple Example

states, and events are represented by arcs. The first symbol in each arc label denotes the event, while the symbol following "/" denotes the corresponding output. Finally, we mark the controllable events by ":u". Thus, in this example, $X = \{0,1,2,3,4\}$, $\Sigma = \{\alpha, \beta, \delta, \mu\}$, $\Gamma = \{\alpha, \beta\}$, and $\Phi = \{\alpha\}$.

There are several basic notions that we will need in our investigation. The first is the notion of liveness. Intuitively, a system is alive if it cannot reach a point at which no event is possible. That is, A is alive if $\forall x \in X$, $d(x) \neq \emptyset$. We will assume that this is the case. A second notion that we need is the composition of two automata, $A_i = (G_i, f_i, d_i, h_i)$ which share some common events. Specifically, let $S = \Sigma_1 \cap \Sigma_2$ and, for simplicity, assume that $\Gamma_1 \cap S = \Gamma_2 \cap S$ (i.e., any shared event observable in one system is also observable in the other). The dynamics of the composition are specified by allowing each automaton to operate as it would in isolation except that when a shared event occurs, it must occur in both systems. Mathematically, we

denote the composition by $A_{12} = A_1 \parallel A_2 = (G_{12}, f_{12}, d_{12}, h_{12})$, where

$$G_{12} = (X_1 \times X_2, \Sigma_1 \cup \Sigma_2, \Gamma_1 \cup \Gamma_2) \tag{2.7}$$

$$f_{12}(x, \sigma) = f_1(x_1, \sigma) \times f_2(x_2, \sigma) \tag{2.8}$$

$$d_{12}(x) = (d_1(x_1) \cap \overline{S}) \cup (d_2(x_2) \cap \overline{S}) \cup (d_1(x_1) \cap d_2(x_2)) \tag{2.9}$$

$$h_{12}(\sigma) = \begin{cases} h_1(\sigma) & \text{if } \sigma \in \Gamma_1 \\ h_2(\sigma) & \text{if } \sigma \in \Gamma_2 \\ \epsilon & \text{otherwise} \end{cases} \tag{2.10}$$

Here we have extended each f_i to all of $\Sigma_1 \cup \Sigma_2$ in the trivial way, namely, $f_i(x_i, \sigma) = x_i$ if $\sigma \notin \Sigma_i$. Note also that h_{12} given by (2.10) is well-defined.

2.2 Stability and Stabilizability

In [5], we define a notion of stability which requires that the trajectories go through a given set E infinitely often:

Definition 2.1 Let E be a specified subset of X. A state $x \in X$ is E-pre-stable if there exists some integer i such that every trajectory starting from x passes through E in at most i transitions. The state $x \in X$ is E-stable if A is alive and every state reachable from x is E-pre-stable. The DEDS is E-stable (respectively, E-pre-stable) if every $x \in X$ is E-stable (respectively, E-pre-stable). ❑

By a cycle, we mean a finite sequence of states $x_1, x_2, \ldots x_k$, with $x_k = x_1$, so that there exists an event sequence s that permits the system to follow this sequence of states. Note that E-stability is equivalent to the absence of cycles that do not pass through E [5]. We also need the following:

Definition 2.2 The radius of A is the length of the longest cycle-free trajectory between any two states of A. The E-radius of an E-stable system A is the maximum number of transitions it takes any trajectory to enter E. ❑

Note that an upper bound on both the radius and the E-radius, for any E, of an E-stable system is n. We refer the reader to [5] for a more complete discussion of this subject and for an $O(n^2)$ test for E-stability of a DEDS. Finally, we note that in [5] and Definition 2.1, we require livenes in order for a system to be stable so that trajectories can be continued indefinitely. While we will continue to require liveness in this paper as we consider compensator design, there are occasions on which it is useful to consider a notion of weak stability, in which all the conditions of Definition 2.1 are met except that A may not be alive. Thus, for a weakly E-stable system, all trajectories pass through E and can only die in E. We note without proof that the algorithm developed in [5] for stability can be used without change to test for weak stability.

In [5], we study stabilization by state feedback. Here, a state feedback law is a map $K : X \to U$ and the resulting closed-loop system is $A_K = (G, f, d_K, h)$ where

$$d_K(x) = (d(x) \cap K(x)) \cup (d(x) \cap \overline{\Phi}) \tag{2.11}$$

Definition 2.3 A state $x \in X$ is E-pre-stabilizable (respectively, E-stabilizable) if there exists a state feedback K such that x is E-pre-stable (respectively, E-stable) in A_K. The DEDS is E-stablilizable if every $x \in X$ is E-stabilizable. ❑

If A is E-stabilizable, then (as we show in [5]), there exists a state feedback K such that every $x \in X$ is E-stable in A_K. We refer the reader to [5] for a more complete discussion of this subject and for an $O(n^3)$ test for E-stabilizability of a DEDS, which also provides a construction for a stabilizing feedback.

2.3 Observability and Observers

In [4], we term a system observable if the current state is known perfectly at intermittent but not necessarily fixed intervals of time. Obviously, a necessary condition for observability is that it is not possible for our DEDS to generate arbitrarily long sequences of unobservable events, i.e., events in $\overline{\Gamma}$, the complement of Γ. A necessary and sufficient condition for checking this is that if we remove the observable events, the resulting automaton $A|\overline{\Gamma} = (G, f, d \cap \overline{\Gamma}, h)$ must be weakly D_O-stable, where D_O is the set of states that only have observable transitions defined, i.e., $D_O = \{x \in X | d(x) \cap \overline{\Gamma} = \emptyset\}$. This is not difficult to check and will be assumed.

Let us now introduce some notation that we will find useful:

- Let $x \rightarrow^s y$ denote the statement that state y is reached from x via the occurence of event sequence s. Also, let $x \rightarrow^* y$ denote that x reaches y in any number of transitions, including none. We also define the reach of x in A as:

$$R(A, x) = \{y \in X | x \rightarrow^* y\} \tag{2.12}$$

- Let

$$Y_0 = \{x \in X | \nexists y \in X, \sigma \in \Sigma, \text{ such that } x \in f(y, \gamma)\} \tag{2.13}$$

$$Y_1 = \{x \in X | \exists y \in X, \gamma \in \Gamma, \text{ such that } x \in f(y, \gamma)\} \tag{2.14}$$

$$Y = Y_0 \cup Y_1 \tag{2.15}$$

Thus, Y is the set of states x such that either there exists an observable transition defined from some state y to x (as captured in Y_1) or x has no transitions defined to it (as captured in Y_0). Let $q = |Y|$.

- Let $L(A, x)$ denote the language generated by A, from the state $x \in X$, i.e., $L(A, x)$ is the set of all possible event trajectories of finite length that can be generated if the system is started from the state x. Also, let $L_f(A, x)$ be the set of strings in $L(A, x)$ that have an observable event as the last event, and let $\overline{L}(A) = \bigcup_{x \in X} L(A, x)$ be the set of all event trajectories that can be generated by A.

- Given $s \in L(A, x)$ such that $s = pr$, p is termed a prefix of s and we use s/p to denote the corresponding suffix r, i.e., the remaining part of s after p is taken out.

In [4], we present a straightforward design of an observer that produces "estimates" of the state of the system after each observation $\gamma[k] \in \Gamma$. Each such estimate is a subset of Y corresponding to the set of possible states into which A transitioned when the last observable event occurred. Mathematically, if we let a function $\hat{\mathbf{x}} : h(\overline{L}(A)) \rightarrow 2^Y$ denote the estimate of the current state given the observed output string $t \in h(\overline{L}(A))$, then

$$\hat{\mathbf{x}}(t) = \{x \in Y | \exists y \in X \text{ and } s \in L_f(A, y) \text{ such that } h(s) = t \text{ and } x \in f(y, s)\} \quad (2.16)$$

The observer, for which the state space is a subset Z of 2^Y, and the events and observable events are both Γ, is a DEDS which realizes this function. Suppose that the present observer estimate is $\hat{x}[k] \in Z$ and that the next observed event is $\gamma[k+1]$. The observer must then account for the possible occurence of one or more unobservable events prior to $\gamma[k+1]$ and then the occurrence of $\gamma[k+1]$:

$$\hat{x}[k+1] = w(\hat{x}[k], \gamma[k+1]) \triangleq \bigcup_{x \in R(A|\Gamma, \hat{x}[k])} f(x, \gamma[k+1]) \quad (2.17)$$

$$\gamma[k+1] \in v(\hat{x}[k]) \triangleq h\left(\bigcup_{x \in R(A|\Gamma, \hat{x}[k])} d(x)\right) \quad (2.18)$$

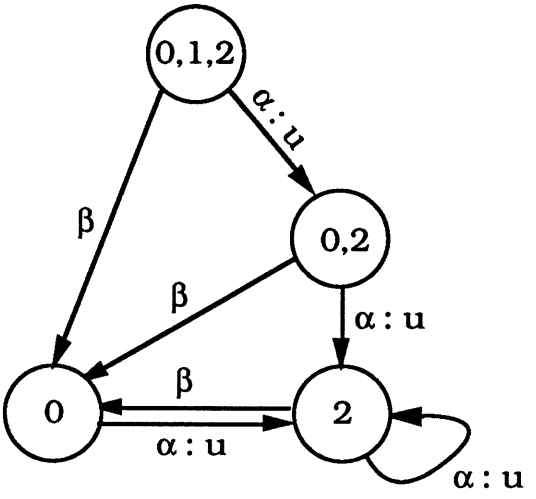

Figure 2.2: Observer for the system in Figure 2.1

The set Z is then in the reach of $\{Y\}$ using these dynamics, i.e., we start the observer in the state corresponding to a complete lack of state knowledge and let it evolve.

Our observer then is the DEDS $O = (F, w, v, i)$, where $F = (Z, \Gamma, \Gamma)$ and i is the identity output function. In some cases, we will treat the observer as a controlled system and discuss stabilizing it. Then, $F = (Z, \Gamma, \Gamma, U)$ and Equation 2.18 becomes

$$\gamma[k+1] \in v(\hat{x}[k]) \triangleq h(\bigcup_{x \in R(A|\Gamma, \hat{x}[k])} (d(x) \cap u[k]) \cup (d(x) \cap \overline{\Phi})) \tag{2.19}$$

The observer for the example in Figure 2.1 is illustrated in Figure 2.2. In [4], we show that a system A is observable iff O stable with respect to its singleton states. We also show that if A is observable then all trajectories from an observer state pass through a singleton state in at most q^2 transitions. Since also there can be at most q singleton states, the radius of the observer is at most q^3. This will play an important role in determining the maximum number of transitions it takes a trajectory from a state, in an output stabilizable system, to pass through E.

2.4 Resiliency

An important aspect of our work is our treatment of resiliency or error recovery. Specifically, suppose that the observed sequence of transitions includes errors corresponding to inserted events, missed events, or mistaken events. We term an observer resilient if after a finite burst of such measurement errors, the observer resumes correct behavior in a finite number of transitions, i.e., the current observer estimate includes the current state of the system. In [4], we construct a resilient observer as follows: The observer O as specified in Equations 2.17 and 2.18 is defined only for event sequences that can actually occur in the system. When measurement error occurs, the resulting observed sequence may not be feasible. In this case, the observer at some point will be in a state such that the next observed event is not defined. In this case, we reset the observer state to $\{Y\}$, i.e., to the condition of knowing nothing about the system state. Thus, for each state in Z and for all events that are not defined at that state, we add a transiton to $\{Y\}$. In particular, we modify w and v as follows:

$$w_R(\hat{x},\gamma) = \begin{cases} w(\hat{x},\gamma) & \text{if } \gamma \in v(\hat{x}) \\ \{Y\} & \text{otherwise} \end{cases} \tag{2.20}$$

$$v_R(\hat{x}) = \Gamma \tag{2.21}$$

and we thus construct the observer $O_R = (F, w_R, v_R, i)$. As before, the initial state of O_R is the state $\{Y\}$. We show in [4] that O_R is a resilient observer if A is observable.

2.5 Effect of State Feedback on Observability

As mentioned in the introduction, we will formulate the output stabilizability problem as a problem of stabilization of the observer by state feedback. Applying state

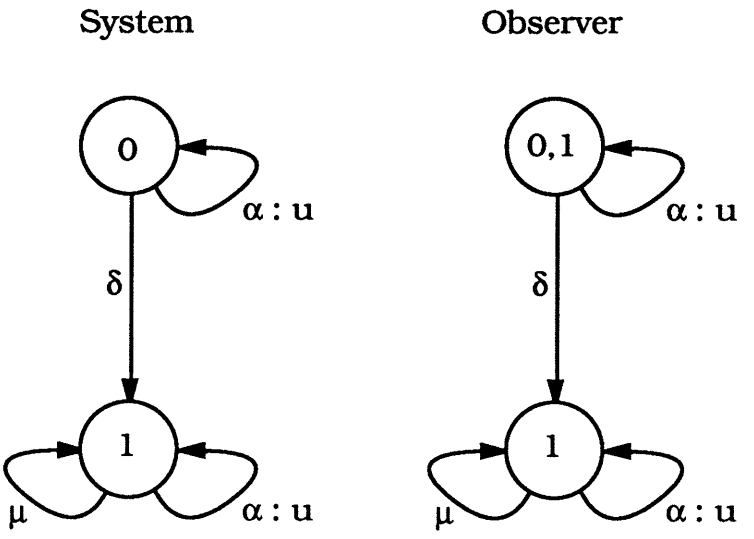

Figure 2.3: Simple Example for Using Control in Observability

feedback to the observer, while preserving liveness, can only enhance observability. In particular, if A is not observable, then it may be possible to find a state feedback for the observer such that the closed loop system is observable. For example, in Figure 2.3, where all the events are observable and α is controllable, if α is disabled at state $\{0,1\}$ of the obsever then the closed loop system is observable and still alive.

2.6 Compensators

We define a compensator as a map $C : \Gamma^* \rightarrow U$. Then, the closed loop system A_C is the same as A but with:

$$\sigma[k+1] \in d_C(x[k], s[k]) \triangleq (d(x[k]) \cap C(h(s[k]))) \cup (d(x) \cap \overline{\Phi}) \qquad (2.22)$$

where $s[k] = \sigma[0] \cdots \sigma[k]$ with $\sigma[0] = \epsilon$: For output stabilizability, we only need to define compensators for strings in $h(\overline{L}(A))$. However, when we talk about resiliency

in Section 5, we need to worry about defining C for arbitrary strings in Γ^*.

One constraint we wish to place on our compensators is that they preserve liveness. Thus, suppose that we have observed the output string s, so that our observer is in $\hat{\mathbf{x}}(s)$ and our control input is $C(s)$. Then, we must make sure that any x reachable from any element of $\hat{\mathbf{x}}(s)$ by unobservable events only is alive under the control input $C(s)$. That is, for all $x \in R(A|\overline{\Gamma}, \hat{\mathbf{x}}(s))$, $d_C(x,s)$ should not be empty. This leads to the following:

Definition 2.4 Given $Q \subset X$, $F \subset \Phi$, F is Q-compatible if for all $x \in R(A|\overline{\Gamma}, Q)$, $(d(x) \cap F) \cup (d(x) \cap \overline{\Phi}) \neq \emptyset$. A compensator C is A-compatible if for all $s \in h(\overline{L}(A))$, $C(s)$ is $\hat{\mathbf{x}}(s)$-compatible. ❑

Suppose that a compensator is such that for all output stings s and t such that the estimate of the current state given s is the same as the estimate given t, the compensator value given s is the same as the value given t. In this case, we can represent C as a cascade of the observer and a map $K : Z \to U$, which can also be thought of as a state feedback for the observer:

Definition 2.5 A compensator C is O-compatible if for all $s, t \in h(\overline{L}(A))$, such that $\hat{\mathbf{x}}(s) = \hat{\mathbf{x}}(t)$, $C(s) = C(t)$. The corresponding map $K : Z \to U$ such that

$$C(s) = K(v(\{Y\}, s))$$

for $s \in h(\overline{L}(A))$, is termed the observer feedback for C. ❑

We will see in Section 3 that we can restrict attention to O-compatible compensators in order to address the stabilization problem.

3 Two Notions of Output Stabilizability

In this section, we present and analyze two notions of output stabilizability. While it certainly is possible for a system to be output stabilizable without being observable (for example, if it is stable), we will, for simplicity, assume observability. Also, while a system must be stabilizable in order to be output stabilizable, we will not explicitly assume stabilizability. Rather, checking stabilizability will be incorporated into our test for output stabilizability.

The obvious notion of output E-stabilizability is the existence of a compensator C so that the closed-loop syatem A_C is E-stable. Because of the intermittent nature of our observations, it is possible that such a stabilizing compensator may exist, so that we are sure that the state goes through E infinitely often, but so that we never know when the state is in E. For this reason, we define a stronger notion of output stabilizability that not only requires that the state pass through E infinitely often but that we regularly know when the state has moved into E. We begin with this latter notion which is easier to analyze.

3.1 Strong Output Stabilizability

The key to our analysis of strong output stabilizability is that we will know that the state is in E if and only if the observer state $\hat{\mathbf{x}}$ is a subset of E:

Definition 3.1 A is strongly output stabilizable if there exists a compensator C and an integer i such that A_C is alive and for all $p \in \overline{L}(A_C)$ such that $|p| \geq i$, there exists a prefix t of p such that $|p/t| \leq i$ and $\hat{\mathbf{x}}(h(t)) \subset E$. We term such a compensator a strongly output stabilizing compensator. ❑

What this definition states is that in addition to keeping the system alive, the compensator C also forces the observer to a state corresponding to a subset of E at intervals of at most i observable transitions. The next result shows that we can restrict attention to observer feedback:

Proposition 3.2 *A is strongly output stabilizable if there exists a state feedback $K : Z \to U$ for the observer such that X_I in $A \parallel O_K$ is E_{OC}-stable, where $X_I = \{(x, \{Y\}) | x \in X\}$ is the set of possible initial states in $A \parallel O_K$ and where $E_{OC} = \{(x, \hat{x}) \in Y \times Z | \hat{x} \subset E\}$ is the set of composite states for which the system is in E and we know that the current state is in E.*

Proof: ($\leftarrow$) Obvious.

($\rightarrow$) If we can find a strongly output stabilizing compensator C that is O-compatible and construct the corresponding observer state feedback K, then X_I is certainly E_{OC}-stable in $A \parallel O_K$.

Let l_i be the set of length i elements of $h(\overline{L}(A))$. Given any strongly output stabilizing compensator C_1 for A, we construct the desired one as follows:

Let $Z_1 = \{\{Y\}\}$ be the set that consists of the initial state $\{Y\}$ of O and let $K(\{Y\}) = C_1(\epsilon)$. Let $S_{11}, \ldots S_{1k_1}$ be a collection of disjoint subsets of l_1 such that (a) $\bigcup_i S_{1i} = l_1$; (b) for all $\sigma \in S_{1i}$, $v(\{Y\}, \sigma) = \hat{x}_i$ for some $\hat{x}_i \in Z$; and (c) for any S_{1i}, S_{1j}, $i \neq j$, $\hat{x}_i \neq \hat{x}_j$. Let us term such a collection of subsets an l_1-collection. For each $\hat{x}_i$ such that $\hat{x}_i \notin Z_1$, pick some $\alpha_i \in S_{1i}$ and let $K(\hat{x}_i) = C_1(\alpha_i)$. Construct a compensator C_2 such that for all output strings of the form σs, for some $\sigma \in S_{1i}$, $C_2(\sigma s) = C_1(\alpha_i s)$. Clearly, C_2 is a strongly output stabilizing compensator for A. Also, let $Z_2 = Z_1 \cup \bigcup_i \hat{x}_i$ which denotes the set of observer states for which we have defined K so far.

We repeat this construction for l_2, l_3, etc. After step $j-1$, C_j is a strongly output stabilizing compensator for A, and we will have defined K for observer states Z_j that can be reached by $\{Y\}$ with output strings of length at most $j-1$. At step j, let $S_{j1}, \ldots S_{jk_j}$ be the l_j-collection. For each $\hat{x}_i$ such that $v(\{Y\}, S_{ji}) = \hat{x}_i$ and $\hat{x}_i \notin Z_j$, pick some $a_i \in S_{ji}$ and let $K(\hat{x}_i) = C_j(a_i)$. Construct a compensator C_{j+1} such that for all output strings of the form ts, for some $t \in S_{ji}$, $C_{j+1}(ts) = C_j(r_i s)$. Clearly, C_{j+1} is a strongly output stabilizing compensator for A. Also, let $Z_{j+1} = Z_j \cup \bigcup_i \hat{x}_i$.

Proceed in this fashion until, at some step j, $Z_j = Z$, which implies that we have defined a feedback for all observer states. The reach of X_I in $A \parallel O_K$ is alive since by construction $K(\hat{x})$ is $\hat{x}$-compatible. Since also C_j is a strongly output stabilizing compensator for A, the compensator C defined by $C(s) = K(v(\{Y\}, s))$ is a strongly output stabilizing compensator for A. Therefore, X_I in $A \parallel O_K$ is E_{OC}-stable. ❑

Since O describes all the behavior that can be generated by A, we have the following which states that it is necessary and sufficient to check the stability of O with respect to the observer states that are subsets of E, while paying attention to keeping the system alive:

Proposition 3.3 *A is strongly output stabilizable iff there exists a state feedback $K : Z \to U$ for the observer such that O_K is stable with respect to $E_O = \{\hat{x} \in Z | \hat{x} \subset E\}$ and for all $\hat{x} \in Z$, $K(\hat{x})$ is $\hat{x}$-compatible. Furthermore, if A is strongly output stabilizable then the trajectories in the reach of X_I in $A \parallel O_K$ go through E_{OC} in at most nq^3 transitions.*

Proof: A straightforward consequence of Proposition 3.2 and the fact that the radius of O is at most q^3. ❑

As an example, consider the system in Figure 3.1, where $E = \{1, 2\}$ and where all events are observable. Note that in this case, we need to check the stabilizability

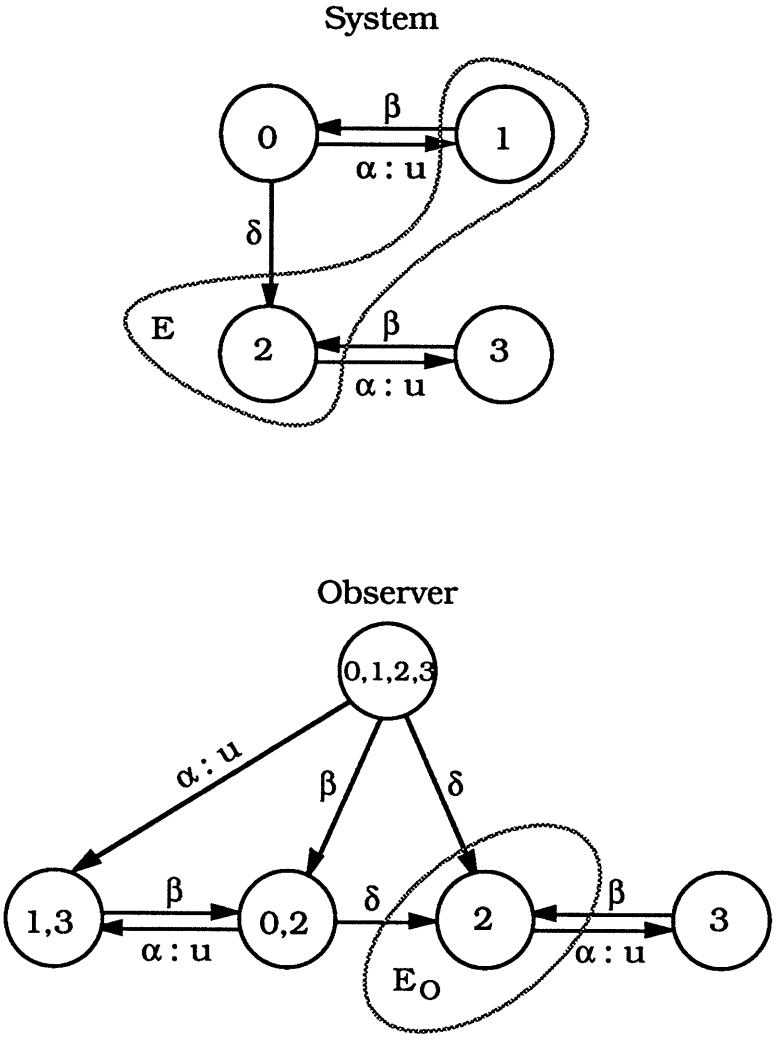

Figure 3.1: Example for Strong Output Stabilizability (all the events are observable)

of the observer with respect to $E_O = \{2\}$. We achieve stability if α is disabled at the observer state $\{0,2\}$. Proposition 3.3 essentially tells us that we can test strong output stabilizability by testing the observer for stabilizability. The following algorithm performs this test and constructs a feedback for strong output stabilization. It is very similar to our algorithm for pre-stabilizability in [4]:

Proposition 3.4 The following algorithm is a test for strong output stabilizability. It has complexity $O(q^3|Z|)$:

Algorithm Let $Z_0 = E_O$ and iterate:

$$\begin{aligned} P_{k+1} &= \{\hat{x} \in Z | \{\gamma \in v(\hat{x}) | w(\hat{x},\gamma) \in P_k\} \text{ is } \hat{x}\text{-compatible}\} \\ K(\hat{x}) &= \{\gamma \in v(\hat{x}) | w(\hat{x},\gamma) \in P_k\} \text{ for } \hat{x} \in P_{k+1} \\ Z_{k+1} &= Z_k \cup P_{k+1} \end{aligned}$$

Terminate when $Z_{k+1} = Z_k = Z^*$. A is strongly output stabilizable iff $Z = Z^*$. The corresponding feedback is K as computed above.

Proof: The proof is straightforward and based on the proof of the algorithm for testing pre-stabilizability in [5]. Computational complexity follows from the fact that the observer has $|Z|$ states and the algorithm terminates in at most q^3 steps. ❑

3.2 Output Stabilizability

In this section, we study the following somewhat weaker notion:

Definition 3.5 A is output stabilizable (respectively, output pre-stabilizable) with respect to E if there exists a compensator C such that A_C is E-stable (respectively, E-pre-stable). We term such a compensator an output stabilizing (respectively, output pre-stabilizing) compensator. ❑

Note that this definition implicitly assumes that there exists an integer i such that the trajectories in A_C go through E in at most i transitions. Using this bound, we can show that output pre-stabilizability and liveness are necessary and sufficient for output stabilizability, as is the case for stabilizability and pre-stabilizability (see [5]):

Proposition 3.6 *A is output stabilizable iff A is output pre-stabilizable while preserving liveness (i.e., the closed loop system is pre-stable and alive).*

Proof: ($\rightarrow$) Obvious.

($\leftarrow$) Let C be an output pre-stabilizing compensator that preserves liveness. Then, for each $x \in X$, there exists an integer i such that the trajectories from x in A_C go though E in at most i transitions. Thanks to our assumption that A cannot generate arbitrarily long sequences of unobservable events, for each $x \in X$, there exists an integer j such that the trajectories from x in A_C go through E in at most j observable transitions. Let j^* be the maximum over all j. Then, we know that the trajectories in A_C go through E in at most j^* observable transitions independently of the initial state. In order to prove our result, we will construct a stabilizing compensator C' using C and j^*. Specifically, given $s \in h(\overline{L}(A_C))$, let s^* denote the suffix of s for which $|s^*| = |s| \bmod j^*$, and let $C'(s) = C(s^*)$. Clearly, $A_{C'}$ is alive. Also, $A_{C'}$ is E-stable since it is guaranteed to go through E at least once every j^* observable transitions. Therefore, A is output stabilizable. ❑

This result shows us that in order to design a stabilizing compensator, we only need to design a pre-stabilizing compensator. Our construction of a pre-stabilizing compensator involves (a) constructing a modified observer *which keeps track of the states the system can be in if the trajectory has not yet passed through* E, (b) formulating the problem of pre-stabilizing A by ouput feedback as a problem of stabilizing this ob-

server by state feedback, and (c) constructing a pre-stabilizing compensator by using this observer and the state feedback constructed in (b).

To provide the motivation behind our approach, consider the system in Figure 3.1. For output stabilizability, we do not really need to disable α (as we had to for strong output stabilizability). Consider the loop in the observer that consists of the states $\{1,3\}$ and $\{0,2\}$. If the system is in state 1 (respectively, state 2), it is already in E. If the system is in state 3 (respectively, state 0), it makes a transition into E after the next event. Therefore, A is stable and thus is trivially output stabilizable (without disabling any event). This example illustrates the key idea in our analysis of output stabilizability: we must keep track of those state trajectories that have not yet passed through E; if that set becomes empty at some point, we will know that the system has passed through E, although we may not know the point in time at which it did.

The following construction allows us to perform this function: Delete all events in A that originate from the states in E and construct the corresponding observer. Let A_E denote this system and let $O_E = (F_E, w_E, v_E)$ denote its observer. For example, Figure 3.2 illustrates such an automaton and observer for the system in Figure 3.1. The observer O_E captures all the behavior of A until its trajectories enter E. When we look at the states of O_E, we see that there are some "trapping" states, each of which is a subset of E and thus has no events defined. Let us consider an event trajectory s in A and the corresponding trajectory $h(s)$ in O_E that starts from the initial state $\{Y\}$. If the trajectory ever evolves to a "trapping" state in O_E, then we know that it has passed through E in A. Other states of O_E may have some elements in E and some elements that are <u>not</u> in E. Let $\hat{x}$ be such a state of O_E, then for a

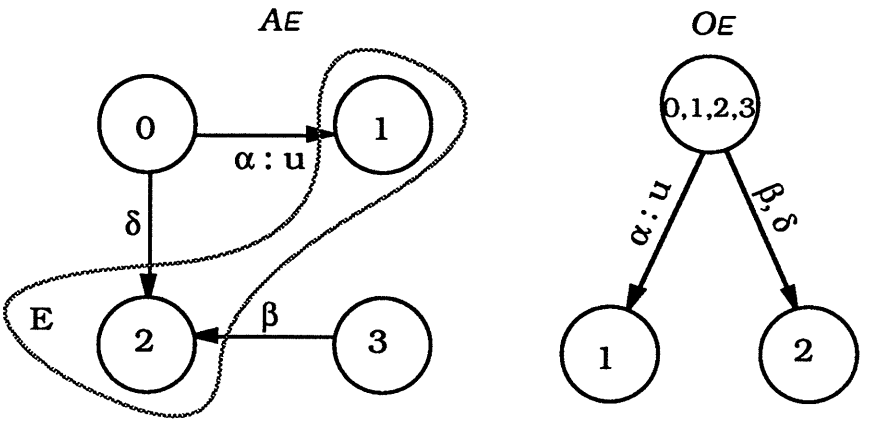

Figure 3.2: Example for A_E and O_E (all the events are observable)

trajectory that evolves to $\hat{x}$, the system can be in one of the states in $\hat{x} \cap \overline{E}$ only if that trajectory has not passed through E yet. Even though O_E keeps track of trajectories that have not passed through E yet, it does not keep track of enough information to design a pre-stabilizing compensator, since, in order to preserve liveness, we also need to know all the states that the system can be in so that we can check if our control input keeps the system alive: The automaton

$$Q = (F_Q, w_Q, v_Q) = O_E \parallel O \tag{3.1}$$

together with the initial state (Y, Y) keeps track of all the information we need for designing an output stabilizing compensator. Note that

$$w_Q((y_1, y_2), \sigma) = (w_E(y_1, \sigma), w(y_2, \sigma)) \tag{3.2}$$

and $v_Q((y_1, y_2)) = v_E(y_1)$. The state space of Q, is $W = R(Q, (Y, Y))$. Figure 3.3 illustrates the automaton Q for the system in Figure 3.1. Note that the number of states of Q is the same as that of O_E. For each state of Q, the second component

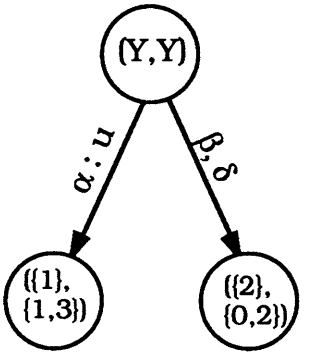

Figure 3.3: Example of the Automaton Q (all the events are observable)

denotes the set of states that the system can be in, whereas the first component denotes the set of states that the system can be in if the trajectory has not gone through E yet.

The following lemma shows that the problem of output pre-stabilization can be formulated as a problem of pre-stabilization of Q. The key is to find a state feedback K for Q, which we can then adapt to a corresponding compensator for A, and which forces all trajectories in Q_K to have finite length. This in turn will force corresponding trajectories in A to go through E in a finite number of transitions. In doing this, however, we need to make sure that the compensator for A keeps A alive:

Lemma 3.7 *A is output pre-stabilizable with respect to E while preserving liveness iff there exists a feedback $K : W \rightarrow U$ such that for all*

$$(y_1, y_2) \in R(Q_K, (Y, Y))$$

$K((y_1, y_2))$ is y_2-compatible, and Q_K is pre-stable with respect to its dead states, i.e., with respect to the states y such that $v_{Q_K}(y) = \emptyset$.

Proof: ($\rightarrow$) Straightforward by assuming the contrary.

($\leftarrow$) We claim that the compensator defined by

$$C(s) = K(w_{Q_K}((Y,Y),s))$$

for $s \in L(Q_K,(Y,Y))$ and $C(s) = \Phi$ for all other s, pre-stabilizes A and we prove this as follows: Thanks to the compatibility condition, A_C is alive. Also,

$$h(\overline{L}(A_C)) \subset L(Q_K,(Y,Y))\Gamma^*$$

Given $s \in \overline{L}(A_C)$, if $s \in L(Q_K,(Y,Y))$ then the trajectory may not have passed through E yet. If $s \notin L(Q_K,(Y,Y))$, suppose that $s = p\sigma$ for some $p \in L(Q_K,(Y,Y))$ and $\sigma \in \Gamma$. Since σ is not defined at $w_{Q_K}((Y,Y),p)$, σ could have occured only if the trajectory has already passed through E. Since also all strings in $L(Q_K,(Y,Y))$ are finite and C preserves liveness, A_C is E-pre-stable. ❑

In order to construct a compensator as proposed by the above lemma, let us first characterize the states in Q that we can "kill" while preserving liveness in A. In particular, let E_Q be the set of states $y = (y_1,y_2) \in W$ so that we can find a y_2-compatible set of events $F \subset \Phi$ which, if used as a control input at y, disables all events defined from y, i.e.,

$$E_Q = \{y = (y_1,y_2) \in W | \exists F \subset \Phi \text{ such that } v_{QF}(y) = \emptyset \text{ and } F \text{ is } y_2\text{-compatible}\} \tag{3.3}$$

where $v_{QF}(y) = (v_Q(y) \cap F) \cup (v_Q(y) \cap \overline{\Phi})$. For example, consider the system in Figure 3.4, where Figure 3.4(a) illustrates A, (b) illustrates A_E, (c) illustrates the observer O for A and (d) illustrates the observer O_E for A_E. The automaton Q for this example is illustrated in Figure 3.5(a). Note that we can disable β at both of the states (2,123) and (2,2) so that no transitions are enabled in Q at these states, but the states 1, 2,

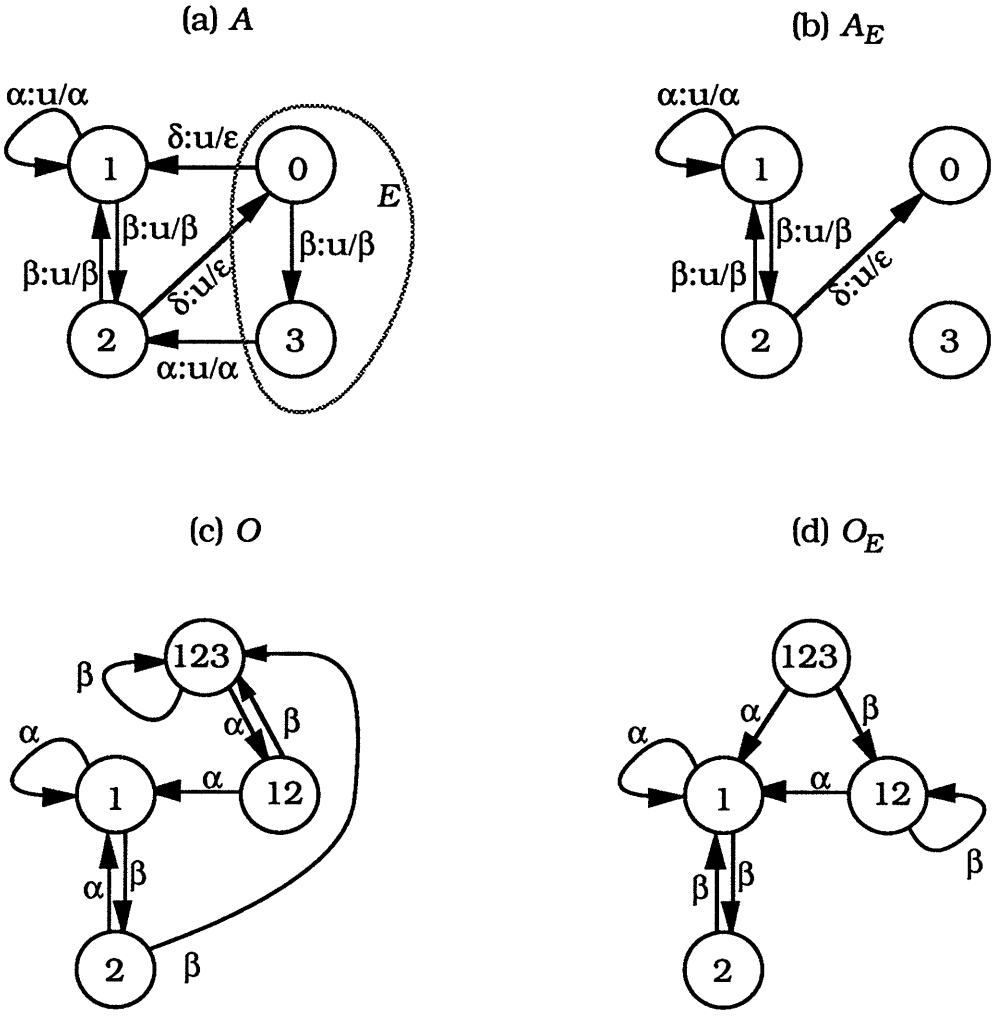

Figure 3.4: Output Stabilizability Example: (a) The system A, (b) A_E, (c) the observer O for A, and (d) the observer O_E for A_E.

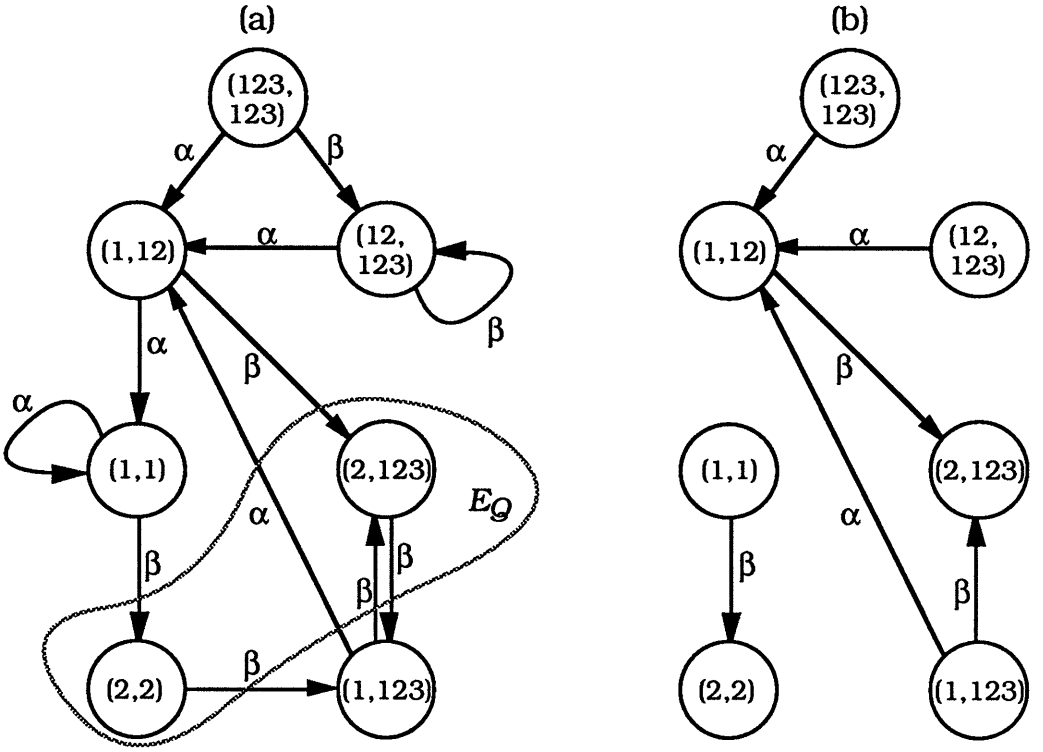

Figure 3.5: Output Pre-stabilization of Figure 3.4 (recall that α and β are both controllable and observable): (a) Automaton Q, and (b) Q_K as computed by Algorithm 3.9.

and 3 remain alive in A. Thus, $E_Q = \{(2,123),(2,2)\}$. Therefore, for this example, if we can find a feedback K so that Q_K is E_Q-pre-stable and alive, then, using Q and this feedback, we can construcut a compensator that pre-stabilizes A, as we did in the proof of Lemma 3.7:

Proposition 3.8 A is ouput pre-stabilizable while preserving liveness iff there exists a state feedback K_0 such that Q_{K_0} is E_Q-pre-stable and for all $(y_1,y_2) \in W$, $K((y_1,y_2))$ is y_2-compatible in A. Furthermore, the compensator defined by

$$C(s) = K(w_{Q_{K_0}}((Y,Y),s))$$

for $s \in L(Q_K,(Y,Y))$ and $C(s) = \Phi$ for all other s, pre-stabilizes A, where

$$K(y=(y_1,y_2)) = \begin{cases} F \subset \Phi | v_{QF}(y) = \emptyset \text{ and } F \text{ is } y_2\text{-compatible} & \text{if } y \in E_Q \\ K_0(y) & \text{otherwise} \end{cases}$$

Finally, the trajectories in A_C go through E in at most nq^3 transitions.

Proof: Straightforward using Lemma 3.7 and the fact that the radius of the observer is at most q^3. ❑

We now present an algorithm to test for output pre-stabilizability and to construct the corresponding feedback by appropriately modifying Algorithm 3.4 for Q:

Proposition 3.9 The following algorithm is a test for output pre-stabilizability while preserving liveness. It has complexity $O(q^3|W|)$:

Algorithm Let $Z_0 = E_Q$ and for $y = (y_1,y_2) \in E_Q$, let $K(y) = F \subset \Phi$ where F is such that $v_{QF}(y) = \emptyset$ and F is y_2-compatible. Iterate:

$$\begin{aligned} P_{k+1} &= \{y \in W | \{\gamma \in v_Q(y) | w_Q(y,\gamma) \in P_k\} \text{ is } y_2\text{-compatible in } A\} \\ K(y) &= \{\gamma \in v_Q(y) | w_Q(y,\gamma) \in P_k\} \text{ for } y \in P_{k+1} \\ Z_{k+1} &= Z_k \cup P_{k+1} \end{aligned}$$

Terminate when $Z_{k+1} = Z_k = Z^*$. A is output pre-stabilizable iff $(Y,Y) \in Z^*$. The corresponding feedback is K as computed above. ❑

Figure 3.5(b) illustrates the closed loop system Q_K after this algorithm is applied to Q in Figure 3.5(a). In order to construct a compensator that pre-stabilizes the system in Figure 3.4(a), we use the range of (123,123) in Q_K as follows: Initially (i.e., before any observable events are seen so that we are in (123,123) of Q_K), we disable β. After α is observed (so that the state in Q_K is (1,12)), α is disabled, while β is enabled, and finally, after β is observed (corresponding to a transition to the state (2,123)), β is disabled while α is enabled. When α occurs again, we know that all the trajectories have passed through E, and thus we do not care about what the control input is after this point as long as it keeps the system alive.

In [5] we have termed a feedback to be maximally restrictive if we cannot disable any other event at any state while preserving liveness. We can generate such a feedback using the algorithm in Proposition 3.9 if we choose $K(y)$ such that removing any event from $K(y)$ violates compatibility. In [5], we have also defined a feedback to be minimally restrictive if, for each state, enabling any event, which is otherwise disabled, violates pre-stability. We have also shown that, a minimally restrictive feedback can be generated from a maximally restrictive one by arbitrarily enabling events (that are otherwise disabled) until pre-stability is violated. In the same manner, we can generate a minimally restrictive feedback from the feedback generated by the algorithm in Proposition 3.9.

We now turn our attention to output stabilizing compensators. Note that if, at some point, we are certain that the trajectory has passed through E then we can force the trajectory to go through E again by starting the compensator over, i.e., by

ignoring all the observations to date and using the pre-stabilizing compensator on the new observations (see the proof of Proposition 3.6). In the proof of Proposition 3.6, we computed an integer j^* so that all the trajectories are guaranteed to go through E in at most j^* transitions independently of the initial state of the system, and so that we can "reset" the output pre-stabilizing compensator after every set of j^* transitions. However, in some cases, it may not be necessary to wait for j^* transitions. In what follows, we present an approach which allows us to detect, as soon as possible, that the trajectory has passed though E.

Given an output pre-stabilizable A, suppose that C is the corresponding compensator and K is the corresponding Q-feedback for C. Recall that for Q_K, no events are defined at states $(y_1, y_2) \in E_Q$, and in general, given some $y = (y_1, y_2) \in R(Q_K, (Y, Y))$, not all events defined at y_2 are defined at y. Given an output trajectory of A_C, let us trace the corresponding trajectory in Q_K starting from the state (Y, Y). Suppose that we observe a transition which is <u>not</u> defined at the current state of Q_K. By the way we have constructed Q_K we know that the occurence of such a transition implies that the trajectory has already passed through E. This is precisely the mechanism which we use to detect that the trajectory has passed through E. So, given $s \in h(\overline{L}(A_C) \cap L(Q_K, (Y, Y))$, let $y = w_{Q_K}((Y, Y), s)$ and suppose that the next observation is a transition $\sigma \notin v_{Q_K}(y)$, and thus we know that the trajectory has passed through E. At this point, we wish to force the trajectory to pass through E again, but in doing so, we can use our knowledge of the set of states that the system can be in at the time we have detected that the trajectory has passed through E, i.e., $w(y_2, \sigma)$. What we would then like to do is to have Q transition to the state $z = (w(y_2, \sigma), w(y_2, \sigma))$. However, as we have defined it so far, z may <u>not</u>

be in W. What we must do in this case is to augment W with all such z's and any new subsequent states that might be visited starting from such a z and using an extension of the dynamics of Q. Specifically, the dynamics of Q given in (3.2) can be defined for arbitrary subsets $y_1, y_2 \subset Y$, as can its restriction w_{Q_K} by feedback. We modify this definition as follows: if $w_{EK}(y_1, \sigma) = \emptyset$, then we set $w_{Q_K}((y_1, y_2), \sigma)$ to $(w(y_2, \sigma), w(y_2, \sigma))$. Let W^a be the union of the reaches of all states of the form (Y', Y') with $Y' \subset Y$ and define $Q^a = (F^a, w, v)$ where $F^a = (W^a, \Gamma, \Gamma)$. Note that $E_Q \subset W^a$ and $R(Q_K, (Y, Y)) \subset W^a$. If in fact any $z = (Y', Y')$ is pre-stabilizable with respect to $R(Q_K, (Y, Y))$ in Q^a, then we can force the trajectory to pass through E. The next result states that pre-stabilizability of Q is sufficient for being able to do this:

Proposition 3.10 If there exists a feedback K for Q such that Q_K is E_Q-pre-stable and $K(y)$ is y_2-compatible, then there exists a feedback K' such that for any $Y' \subset Y$, $z = (Y', Y')$ is pre-stable with respect to $R(Q_K, (Y, Y))$ in $Q^a_{K'}$ and $K'(y)$ is y_2-compatible for each $y = (y_1, y_2) \in R(Q^a_{K'}, z)$.

Proof: Straightforward by assuming the contrary. ❑

Note that K' can be chosen so that $K'(y) = K(y)$ for all $y \in R(Q_K, (Y, Y))$ and the algorithm in Proposition 3.9 can be used for constructing such a K'.

In order to construct an output stabilizing compensator, we use the above proposition recursively as follows: Let K_0 be a feedback that pre-stabilizes Q and preserves liveness, as can be constructed using the algorithm in Proposition 3.9. Let Z_0 represent the initial state of Q_{K_0} and let W_0 represent the range of Z_0, i.e., the states we may be in when we know that the trajectory has already passed through E:

$$Z_0 = (Y, Y) \tag{3.4}$$

$$W_0 = R(Q_{K_0}, Z_0) \tag{3.5}$$

We then augment Z_0 to include the states to which we may "reset" our compensator, i.e.,

$$Z_1 = Z_0 \cup \{(\hat{x}, \hat{x}) | \hat{x} = w(y_2, \sigma) \text{ for some } y = (y_1, y_2) \in W_0 \text{ and } \sigma \in \hat{v}(y_2, K_0(y))\} \tag{3.6}$$

where $\hat{v}(y_2, K_0(y)) = (v(y_2) \cap K_0(y)) \cup (v(y_2) \cap \overline{\Phi})$. Next, we find a feedback K_1 that satisfies Proposition 3.10 for each $(Y', Y') \in Z_1$. Finally, we let $W_1 = R(Q_{K_1}, Z_1)$. Proceeding in this fashion, we construct W_2, W_3, etc., until $W_{k+1} = W_k = W'$ for some k (note that k must necessarily be finite). Let K' be the corresponding feedback, then

- $Q_{K'}$ is E_Q-pre-stable,
- $K'(y)$ is y_2-compatible for all $y \in W'$, and
- for all $y \in E_Q \cap W'$ and $\sigma \in \hat{v}(y_2, K'(y))$,

$$(w(y_2, \sigma), w(y_2, \sigma)) \in W'$$

Finally, we construct an automaton $Q' = (F', w', v')$ where $F' = (W', \Gamma, \Gamma)$ which includes the transitions to states in Z', i.e.,

$$w'(y, \sigma) = \begin{cases} w_Q(y, \sigma) & \text{if } \sigma \in v_{Q_{K'}}(y) \\ (w(y_2, \sigma), w(y_2, \sigma)) & \text{otherwise} \end{cases} \tag{3.7}$$

$$v'(y) = \hat{v}(y_2, K(y)) \tag{3.8}$$

Then, the compensator defined by

$$C(s) = K'(w'((Y, Y), s)) \tag{3.9}$$

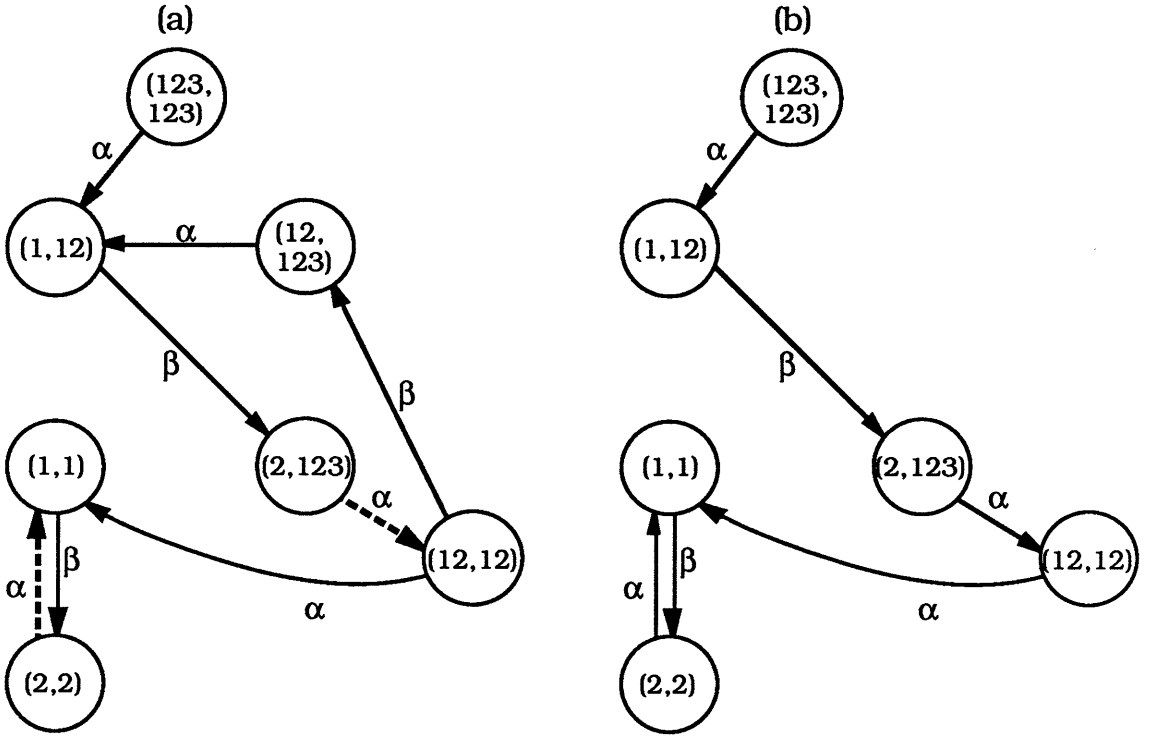

Figure 3.6: Output Stabilization of Figure 3.4(recall that both α and β are controllable: (a) Adding the new states (through the dashed arcs), (b) Q'.

for all $s \in L(Q', (Y, Y))$ stabilizes A. Thus the compensator consists of the automaton Q', started in (Y, Y) and the feedback $K' : W' \rightarrow 2^{\Phi}$ so that the desired compensator is given by the Equation (3.9). For example, for the system in Figure 3.4, we need to pre-stabilize the state (12,12) (see Figure 3.6(a)). The resulting automaton Q' that produces the desired compensator is shown in Figure 3.6(b).

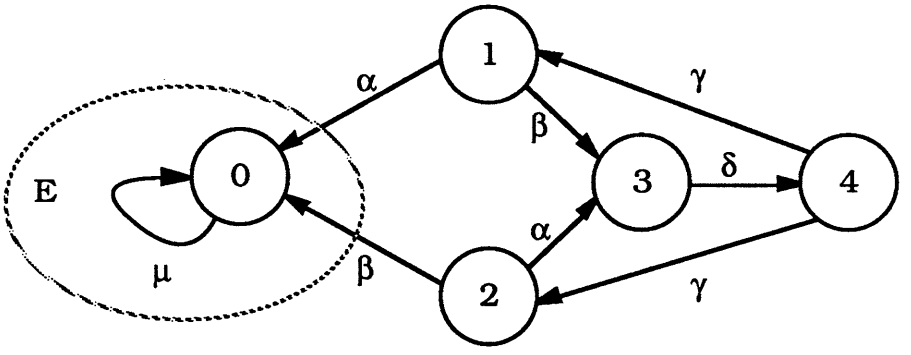

Figure 4.1: Stabilizable, Observable, But Not Output Stabilizable System (all the events are controllable and observable)

4 Sufficient Conditions Testable in Polynomial Time

The previous section presented necessary and sufficient conditions for output stabilizability that can be tested in polynomial time in the cardinality of the state space of the observer O (note that the cardinality of the state space of Q is polynomial in the cardinality of the state space of O). However, while in many cases the observer state space may be sufficiently compact, there are worst cases in which the cardinality of the state space of O is exponential in q (see [4]). In this section, we present sufficient conditions that can always be tested in polynomial time in q.

It is well known in linear sytem theory that controllability and observability imply stabilizability using dynamic output feedback. Unfortunately, stabilizability and observability do not imply output stabilizability in our framework. For example, consider the system in Figure 4.1, where all the events are controllable and observable. This system is stabilizable by disabling β at state 1 and α at 2, and it is also observable. However, it is not output stabilizable, since we can never distinguish between states 1 and 2, and thus we cannot selectively disable α or β.

The reason for this phenomenon is that our notion of observability is much weaker than the corresponding system theory notion, since we only require that the state is known intermittently. We start this section by showing that a result similar to that in system theory can be achieved if we assume that after a finite number of transitions, and for each transition after that, we have perfect knowledge of the current state (this condition is equivalent to the notion of observability of Ramadge [6]). Later in this section, we also show how this condition may sometimes be satisfied by choice of feedback. Finally, we present a weaker sufficient condition based on a notion of always observability that we have defined in [4].

To formalize the first sufficient condition, we need the following notion of transition-function-invariance that we have defined in [5]: Given A and $Q \subset X$, Q is f-invariant in A if all state trajectories from Q stay in Q. In [5], we also show that a maximal f-invariant subset of a given set exists and we present an algorithm that computes it. Let E_w be the maximal w-invariant subset of the set of singleton states of O. If $E_w = \emptyset$ and if O is E_w-stable, then at some finite point the observer state will enter E_w and never leave, so that the state will be known perfectly from that point on.

Proposition 4.1 Suppose that (i) $E \cap E_w = \emptyset$; (ii) A is $E \cap E_w$-stabilizable; (iii) O is E_w-stable, then A is output-stabilizable.

Proof: Let K be a state feedback such that A_K is $E \cap E_w$-stable. We then construct a feedback $\hat{K}$ on O by applying K only when the observer state has moved into E_w, i.e.,

$$\hat{K}(\hat{x}) = \begin{cases} K(x) & \text{if } \hat{x} = \{x\} \in E_w \\ \Phi & \text{otherwise} \end{cases}$$

This feedback clearly stabilizes A, and thus, A is output stabilizable. ❑

As an example, consider the system in Figure 2.1 where $E = \{0\}$. Note that $E_w = \{0, 2\}$, $E \cap E_w = \{0\}$, and the observer, illustrated in Figure 2.2 is E_w-stable. A $E \cap E_w$-stabilizing feedback is one that disables α at state 2. Thus, an output stabilizing feedback is one that disables α when the observer estimate is $\{2\}$.

To show that the computational complexity of testing Proposition 4.1 is polynomial in q, we proceed as we did in [4] for testing observability. First, we construct an automaton A' over Y by appropriately eliminating transitions that are not observable, i.e., this automaton models the state transition behavior sampled at the times at which observable events occur. Thus:

$$A' = (G', f', d', i) \tag{4.1}$$

$$G' = (Y, \Gamma, \Gamma, U) \tag{4.2}$$

$$f'(y, \gamma) = f(R(A|\overline{\Gamma}, y), \gamma) \tag{4.3}$$

$$d'(y) = \bigcup_{x \in R(A|\overline{\Gamma}, y)} h(d(x)) \tag{4.4}$$

and the output function is identity. Note also that the observers for A and A' are identical. Next, we construct an automaton that captures the ambiguity in the current state of the system. Let $P = Y \times Y$ and construct the pair automaton O_P with state space P and event set Γ such that

$$w_P(p = (x, y), \gamma) = (f'(x, \gamma) \cup f'(y, \gamma)) \times (f'(x, \gamma) \cup f'(y, \gamma)) \tag{4.5}$$

$$v_P(p) = d'(x) \cup d'(y) \tag{4.6}$$

where $p = (x, y) \in P$ and $\gamma \in \Gamma$. For example, the corresponding automaton O_P for the system in Figure 2.1 is illustrated in Figure 4.2.

As developed in [4], although O_P is a nondeterministic automaton and therefore is certainly not an observer for A, O_P can be used to check the observability of A, or

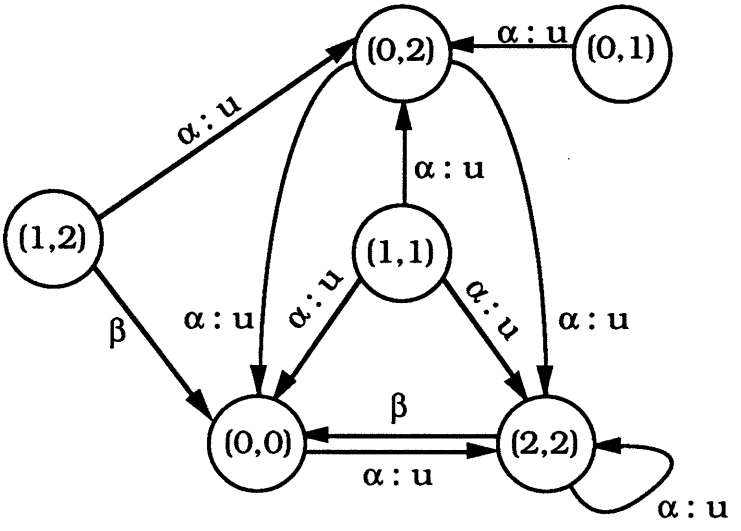

Figure 4.2: Example for the Automaton O_P

equivalently A'. Specifically, the dynamics of O_P have the following interpretation. Suppose that the system might be in either state x or state y, and suppose that the event γ occurs. Then, the next state of A' could be any element of

$$S = f'(x,\gamma) \cup f'(y,\gamma) \tag{4.7}$$

The pair automaton dynamics captures this possible ambiguity by moving from (x,y) to any (x',y') with $x',y' \in S$. Also, there are some special states in O_P, namely those in $E_P = \{(x,x)|x \in Y\}$, corresponding to no ambiguity. It is not difficult to see that observability of A is then equivalent to the E_P-stability of O_P. Similarly, if a set of states of O_P of the form (x,x) is w_P-invariant, then the corresponding set of states in the observer is w-invariant. Thus, we can compute E_w using O_P: We first find V_P, the maximal w_P-invariant subset of E_P, which will be of the form $\{(x,x)|x \in Y'\}$ for some $Y' \subset Y$. It then follows that $E_w = \{\{x\}|x \in Y'\}$:

Proposition 4.2 E_w is the maximal w-invariant subset of the singleton states of O iff $\{(x,x)|\{x\} \in E_w\}$ is the maximal w_P-invariant subset of E_P in O_P.

Proof: Straightforward by assuming contrary in each direction. ❑

As an example, compare Figure 2.2 and Figure 4.2.

Furhermore, it follows from the work we did in [4] that O is E_w-stable iff O_P is $\{(x,x)|\{x\} \in E_w\}$-stable. Since testing a system for stability is equivalent to testing a system for pre-stability (see [5]) which takes quadratic time in the number of states in the sytem, Proposition 4.1 can be tested in $O(q^4)$ time.

If the conditions of Proposition 4.1 are not satisfied, we can test a weaker sufficient condition for output stabilizability while keeping polynomial complexity. Instead of the maximal w-invariant subset of the singleton states, we can use a notion of achieving invariance using state feedback, that we have defined in [5]: Given A and $Q \subset X$, Q is sustainably (f,u)-invariant in A if there exists a state feedback such that Q is alive and f-invariant in the closed loop system. In [5], we also show that a maximal sustainable (f,u)-invariant subset of a given set exists and we present an algorithm that computes it. Let E_u be the maximal sustainable (w,u)-invariant subset of the singleton states and let K_u be the associated state feedback. Note that K_u only needs to act on the singleton states, and thus it can also be thought of as a feedback for A. Note also that K_u needs to disable those events that take states in E_u outside of E_u, and it is unique provided that it only disables such events. As before, if A_{K_u} is $E \cap E_u$-stabilizable and O is E_u-stable, then A is output stabilizable:

Proposition 4.3 Suppose that (i) $E \cap E_u = \emptyset$; (ii) A is $E \cap E_u$-stabilizable; and (iii) O

is E_u-stable. Then if $K_s(x)$ is a stabilizing feedback, the feedback

$$\hat{K}(\hat{x}) = \begin{cases} K_u(x) \cap K_s(x) & \text{if } \hat{x} = \{x\} \in E_u \\ \Phi & \text{otherwise} \end{cases} \tag{4.8}$$

is an output stabilizing feedback for A.

Proof: Straightforward. ❑

As an example, in Figure 4.3, where all events are observable, $E_w = \emptyset$, but $E_u = \{\{0\}, \{2\}\}$ and the associated feedback disables α when the observer is in state $\{0\}$. Furthermore, $E \cap E_u = \{0\}$ and if we disable α at state 2 then we can stabilize A with respect to state 0. Finally, note that O is E_u-stable. Thus, A is output stabilizable, and an output stabilizing feedback is one that disables α when the observer estimate is 0 or 2.

This sufficient condition can also be tested in polynomial time since, similar to Proposition 4.2, E_u is the maximal sustainable (w, u)-invariant subset of the singleton states of O iff $\{(x, x) | x \in E_u\}$ is the maximal sustainable (w_P, u)-invariant subset of E_P in O_P. Furhermore, O is E_u-stable iff O_P is $\{(x, x) | \{x\} \in E_u\}$-stable. Therefore, this sufficient condition for output stabilizability can also be tested in $O(q^4)$ time.

We conclude this section by presenting an even weaker sufficient condition that can also be tested in polynomial time. This condition is based on a notion of always observability that we define in [4]: We term a state x always observable if whenever the system is in x, the observer estimate is $\{x\}$. We term a system a-observable if it is stable with respect to its always observable states. Suppose that A is a-observable and let us construct the automaton A_a which is the same as A except that only events in always observable states can be controllable, i.e., $e_a(x) = d(x)$ for all states x that are not always observable. If A_a is stabilizable then A is also output stabilizable

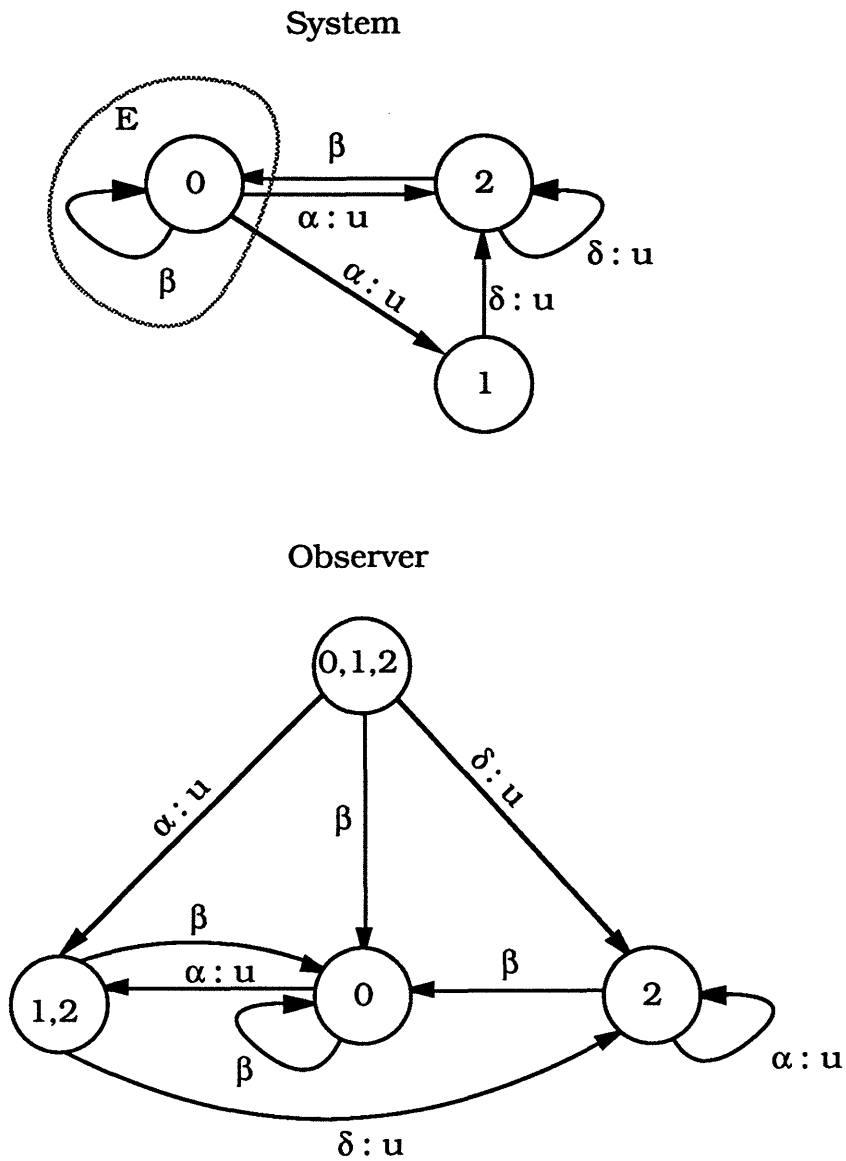

Figure 4.3: Simple Example for Using Control in Observer (all the events are observable)

since whenever we need to exercise control, we have prefect knowledge of the state and thus we can simply use the feedback that stabilizes A_a on those singleton states of the observer that are always observable:

Proposition 4.4 Given an a-observable system A, if A_a is E-stabilizable then A is output stabilizable. ❑

As we show in [4], a-observability can be tested in $O(q^4)$ time, and thus this sufficient condition can also be tested in $O(q^4)$ time.

5 Resiliency

As we did with observability in [4], we can address a problem of robustness. Specifically, in this section we study the property of resilient output stabilizability in the sense that in spite of a burst of observation errors, the system stays alive and goes through E infinitely often.

In order to define what we mean by a resilient stabilizability, we also need to define a notion to represent the discrepancy between two strings. Since the actual point that the burst ends is important for our definition of resiliency, we compare two strings from their beginning and we represent their discrepancy by how much they differ at the end. In particular, we say that the discrepancy between two strings s and t is of length at most i, denoted by

$$\xi(s,t) \leq i \tag{5.1}$$

if there exists a prefix, p, of s and t such that $|s/p| \leq i$ and $|t/p| \leq i$.

Definition 5.1 Given a strongly output stabilizable A, A is resiliently, strongly output stabilizable if there exists a strongly output stabilizing compensator $C : \Gamma^* \rightarrow U$ and an integer i such that for all strings s that can be generated by A_C, i.e.,

- $\forall x \in X$,
- $\forall s \in L_f(A_C, x)$,

for all possible ouput strings t which can be generated by corrupting $h(s)$ with a finite length burst, i.e.,

- $\forall$ positive integers i,

- $\forall t \in \Gamma^*$ such that $\xi(t, h(s)) \leq i$,

the compensator acting on such corrupted strings still strongly stabilizes the system after the error burst has ended. That is, for each such x, s, and t, the compensator $C'(h(s')) \triangleq C(th(s'))$, defined for $s' \in h(L(A, f(x,s)))$ is such that

- the range of $f(x,s)$ is alive in $A_{C'}$, i.e., for all $x \in R(A_{C'}, f(x,s))$, $d_{C'}(x) \neq \emptyset$, and
- for all $p \in L(A_{C'}, f(x,s))$ such that $|p| \geq i$, there exists a prefix p' of p such that $|p/p'| \leq i$ and $f(x, sp) \subset w_{CR}(\{Y\}, th(p')) \subset E$, where w_{CR} is the transition function of the resilient observer O_{CR} for A_C.

We say that C is a resiliently, strongly stabilizing compensator for A. ❑

In the above definition, the requirements on C' ensure that the compensator C acting on the corrupted output string (a) preserves liveness (as stated in the first bullet), and (b) stabilizes A following the burst (as stated in the second bullet).

Let us return to the characterization of strong output stabilizability in Proposition 3.3, but note that we can no longer use O as a basis for constructing a stabilizing compensator since the burst may be an arbitrary string in Γ^*. Therefore, as we did for resilient observability in [4] and explained in Section 2, we will use O_R. In particular, given the observer O and an observer feedback K, define $O_{KR} = (F_{KR}, w_{KR}, v_{KR})$ so that

$$w_{KR}(\hat{x}, \gamma) = \begin{cases} w_K(\hat{x}, \gamma) & \text{if } \gamma \in v_K(\hat{x}) \\ \{Y\} & \text{otherwise} \end{cases} \tag{5.2}$$

$$v_{KR}(\hat{x}) = \Gamma \tag{5.3}$$

We can then define a compensator $C(s) = K(w_{KR}(\{Y\}, s))$ for all $s \in \Gamma^*$. If an error burst now occurs, it may put the system and observer in arbitrary states not necessarily within the reach of the initial states X_I defined in Proposition 3.3. As the following result shows, we can characterize resilient output stabilizability as the stablity of $A \parallel O_{KR}$ for some observer feedback K. In fact, since $A \parallel O_{KR} = A \parallel O_K$, we can use $A \parallel O_K$ instead:

Proposition 5.2 *A is resiliently, strongly output stabilizable if there exists a state feedback $K : Z \rightarrow U$ for the observer such that $A \parallel O_K$ is E_{OC}-stable.*

Proof: ($\rightarrow$) Straightforward by assuming the contrary.

($\leftarrow$) Straightforward since then $C(s) = K(w_{KR}(\{Y, \}, s))$ resiliently, strongly stabilizes A. ❑

Finally, we have the following companion of Proposition 3.2 which states that it is necessary and sufficient to test O for E_O-stability, but since the burst may put the system and the observer in arbitrary states, we need to use X-compatible feedback, in order to preserve liveness:

Proposition 5.3 *A is resiliently, strongly output stabilizable with respect to E iff there exists a state feedback K for the observer such that O_K is E_O-stable and for all $\hat{x} \in Z$, $K(\hat{x})$ is X-compatible.*

Proof: ($\rightarrow$) Assume contrary, then for each K such that O_K is E_O-stable, there exists some $\hat{x} \in Z$ and $x \in Y$ such that $(d(x) \cap K(\hat{x})) \cup e(x) = \emptyset$. Let s be a string such that $\hat{x} = w(\{Y\}, s)$. Suppose that the system started in state x and although no event has occured, the observer observed a burst s. Then, while the system is still in x, the observer is in $\hat{x}$ and no other transition can occur. Therefore, A cannot be resiliently,

strongly output stabilizable and we establish a contradiction.

($\leftarrow$) Straightforward. ❑

An algorithm for testing resilient, strong output stabilizability and constructing a feedback is identical to Algorithm 3.4 except that when we search for a feedback, we search for one that is X-compatible, as opposed to $\hat{x}$-compatible, and the computational complexity is again $O(q^3|Z|)$. Thus, if we can find K that satisfies Proposition 5.3, then $C(s) = K(w_{KR}(\{Y,\}, s))$ is a resiliently, strongly stabilizing compensator for A.

We define resilient output stabilizability similarly:

Definition 5.4 Given output stabilizable A, A is resiliently output stabilizable if there exists an output stabilizing compensator C such that for all strings s that can be generated by A_C, i.e.,

- $\forall x \in X$,
- $\forall s \in L_f(A_C, x)$,

for all possible ouput strings t which can be generated by corrupting $h(s)$ with a finite length burst, i.e.,

- $\forall$ positive integers i,
- $\forall t \in \Gamma^*$ such that $\xi(t, h(s)) \leq i$,

the trajectories starting from $f(x,s)$ visit E infinitely often, i.e., $f(x,s)$ is E-stable in $A_{C'}$, where

$$C'(h(s')) = C(th(s'))$$

for all $s' \in h(L(A, f(x,s)))$. We say that C is a resiliently stabilizing compensator for A.

❑

The following result immediately follows from this definition:

Lemma 5.5 If C is a resilient output stabilizing compensator then $C(s)$ is X-compatible for all $s \in \overline{L}(A)$. ❑

Similar to resilient strong output stabilizability, necessary and sufficient conditions for resilient output stabilizability parallel those of output stabilizability except that we need to use X-compatible feedback. Since, a resilient output stabilizing compensator needs to be defined for all strings in Γ^*, given a feedback K for the automaton Q defined in Section 3.2, we define $Q_{KR} = (G_{KR}, w_{KR}, v_{KR})$ so that

$$w_{KR}(y,\gamma) = \begin{cases} w_{Q_K}(y,\gamma) & \text{if } \gamma \in v_{Q_K}(y) \\ (Y,Y) & \text{otherwise} \end{cases} \tag{5.4}$$

$$v_{KR}(y) = \Gamma \tag{5.5}$$

We can then define a compensator $C(s) = K(w_{KR}((Y,Y),s))$ for all $s \in \Gamma^*$. We state the following companion of Proposition 3.8 where

$$E_{QR} = \{y = (y_1, y_2) \in W | \exists F \subset \Phi \text{ such that } v_{QF}(y) = \emptyset \text{ and } F \text{ is } X\text{-compatible}\} \tag{5.6}$$

Proposition 5.6 A is resiliently ouput stabilizable iff there exists a state feedback K such that Q_K is E_Q-pre-stable and for all $y \in W$, $K(y)$ is X-compatible in A. Furthermore, the compensator defined by

$$C(s) = K(w_{KR}((Y,Y),s))$$

for all $s \in \Gamma^*$ resiliently stabilizes A.

Proof: ($\rightarrow$) Clearly, a feedback K which pre-stabilizes Q exists. By Lemma 5.5, the second condition is satisfied.

($\leftarrow$) Straightforward ❑

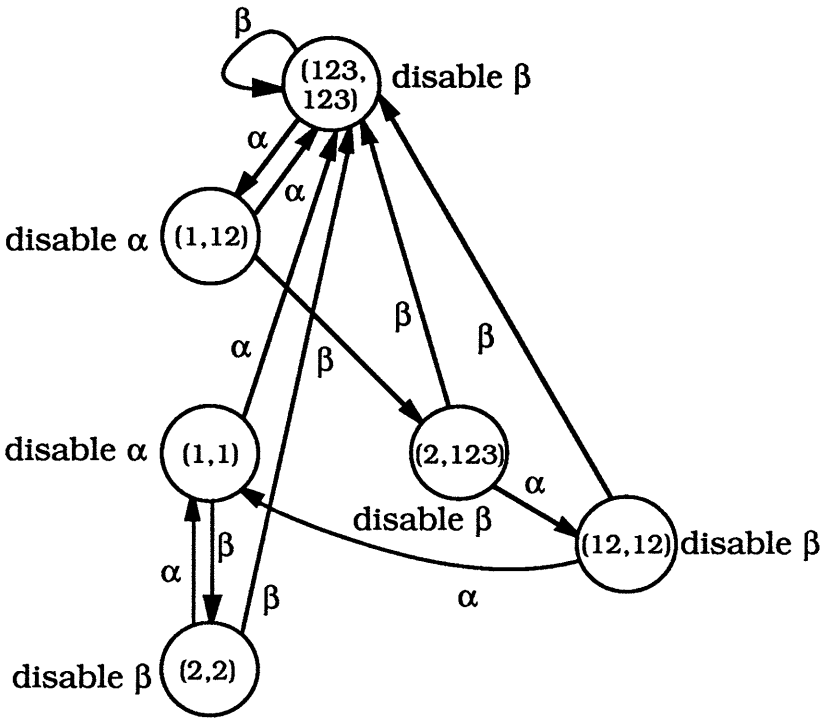

Figure 5.1: Resilient Output Stabilizing Compensator for Figure 3.4

An algorithm for testing resilient output stabilizability and constructing a feedback can be generated from Algorithm 3.4 in a straightforward fashion. In particular, we use E_{QR} in place of E_Q in Algorithm 3.4 and we check X-compatibility, instead of y_2-compatibility.

For example, the feedback we computed for Q in order to stabilize the system in Figure 3.4 is also X-compatible (see Figure 3.6(b)), since, in this case, disabling either, but only one of, α or β does not disable all the events in any state of the system. A resilient output stabilizing compensator for the system in Figure 3.4 is illustrated in Figure 5.1 for which the initial state is (123,123).

6 Conclusions

In this paper, we have introduced notions of output stabilizability and resiliency for discrete-event systems described by finite-state automata, and we have developed algorithms to test for output stabilizablity, resiliency, and to construct resilient output stabilizing compensators. These algorithms are polynomial in the cardinality of the state space of the observer. We have also presented sufficient conditions which can be tested in polynomial time in the cardinality of the state space of the system.

The results presented in this paper provide us with methods for stabilizing DEDS and for ensuring robustsness to observation errors so that catastrophic error propogation is avoided. They also provide the basis for our work in controlling a DEDS so that particular sets of desired strings are tracked. In a subsequent paper, we address this problem and formulate it as the stabilization of the composite of A and an automaton which generates the string or the set of strings that we wish the system to track. Using the results in this paper, we can, in a straightforward way, also address tracking problems in the case of partial observations and observation errors.

References

[1] R. Cieslak, C. Desclaux, A. Fawaz, and P. Variaya. Modeling and control of discrete event systems. In *Proceedings of CDC*, December 1986.

[2] F. Lin and W. M. Wonham. Decentralized supervisory control of discrete event systems. Systems Control Group Report 8612, University of Toronto, July 1986.

[3] J. S. Ostroff and W. M. Wonham. A temporal logic approach to real time control. In *Proceedings of CDC*, December 1985.

[4] C. M. Özveren and A. S. Willsky. Observability of discrete event dynamic systems. Submitted to the *IEEE Transactions on Automatic Control.*

[5] C. M. Özveren, A. S. Willsky, and P. J. Antsaklis. Stability and stabilizability of discrete event dynamic systems. Submitted to the *Journal of the ACM.*

[6] P. J. Ramadge. Observability of discrete event systems. In *Proceedings of CDC*, December 1986.

[7] P. J. Ramadge and W. M. Wonham. Modular feedback logic for discrete event systems. *SIAM J. of Cont. and Opt.*, September 1987.

[8] P. J. Ramadge and W. M. Wonham. Supervisory control of a class of discrete event processes. *SIAM J. of Cont. and Opt.*, January 1987.

[9] J. N. Tsitsiklis. On the control of discrete event dynamical systems. In *Proceedings of CDC*, December 1987.

[10] A. F. Vaz and W. M. Wonham. On supervisor reduction in discrete event systems. *International Journal of Control*, 1986.